UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/829,668 | 04/22/2004 | Douglas C. Burger | 119832-163869 | 6831 |

60172 7590 03/24/2009
SCHWABE, WILLIAMSON & WYATT, P.C.
1420 FIFTH, SUITE 3010
SEATTLE, WA 98101

| EXAMINER | |
|---|---|
| FENNEMA, ROBERT E | |
| ART UNIT | PAPER NUMBER |
| 2183 | |
| MAIL DATE | DELIVERY MODE |
| 03/24/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| | | |
|---|---|---|
| ***Office Action Summary*** | **Application No.**<br>10/829,668 | **Applicant(s)**<br>BURGER ET AL. |
| | **Examiner**<br>ROBERT E. FENNEMA | **Art Unit**<br>2183 |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *26 January 2009*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *37-44,46-50 and 52-60* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *37-44, 46-50, and 52-60* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

## DETAILED ACTION

1. Claims 37-44, 46-50, and 52-60 are pending. Claims 37, 43-44, 46-47, 50, and 52-55 amended as per Applicant's request. Claims 56-60 added as per Applicant's request. Claims 45 and 51 cancelled as per Applicant's request.

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 1/26/2009 has been entered.

### ***Claim Objections***

3. In Claim 46, Line 4, Examiner has interpreted “spanning the another or fame” to read as “spanning the another or same”. However, correction or clarification is required.

4. In Claim 47, Line 4, “instructions” is misspelled. Correction is required.

5. In Claim 55, Line 2, Examiner believes that “includes” should precede instructions. Correction or clarification is required.

6. In Claim 56, Line 1, “executing” is misspelled. Correction is required.

### *Claim Rejections - 35 USC § 102*

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 37-39, 43-44, 46-48, 50, and 52-60 are rejected under 35 U.S.C. 102(b) as being anticipated by Requa et al. ("The Piecewise Data Flow Architecture: Architectural Concepts", herein Requa).

9. As per Claim 37, Requa teaches: A method, comprising:

partitioning a program into a plurality of groups of instructions (Page 426, first column, second paragraph, instructions are grouped into blocks);

assigning a group of instructions selected from the plurality of groups of instructions partitioned from the program to a plurality of interconnected preselected computation nodes (Page 426, first column, second paragraph, the blocks are sent to processors);

loading a subset of instructions of the assigned group of instructions into a frame of buffers spanning the plurality of interconnected preselected computation nodes having been assigned the group of instructions (Page 426, first column, second paragraph, also see Page 433, "The PDF Block Processor". Specifically, see Page 434, Figure 9. the Instruction List, as it is used by all processors, and is a buffer, it is a buffer which spans all the connected nodes/processor); and

executing the subset of instructions as each one of the instructions in the subset of instructions loaded into the frame of spanning buffers receives all necessary associated operands for execution (Page 433, first column, third paragraph, an instruction waits for input operands, then is executed. Also see Page 435, Instruction Issue section. When the instruction in the list has all required operands, it is sent to the instruction-issue section, and then a processor).

10. As per Claim 38, Requa teaches: The method of claim 37, wherein at least one computation node included in the plurality of interconnected preselected computation nodes has at least one input port capable of being coupled to at least one preselected first other computation node included in the plurality of interconnected preselected computation nodes (Figure 1),

the input port to receive input data (Page 427, second column, Paragraph 2 (any consumer can receive any data), also see Page 433, The PDF Block Processor (herein referred to as PDF for the remainder of this claim, as this section will be referenced several times). Additionally, one would recognize a processor/execution unit inherently requires an input port to receive data),

a first store coupled to the at least one input port to store the input data (PDF, paragraph 1, input operands are stored in registers), a second store coupled to an instruction sequencer, the second store to receive and store the at least one instruction (PDF, Paragraph 1, the instruction issue section holds instructions prior to execution),

an instruction wakeup unit to match the input data to the at least one instruction (PDF, paragraph 1, the operand source fields are modified as data comes in), at least one execution unit to execute the at least one instruction using the input data to produce output data (PDF, paragraph 1, the instructions are executed after receiving inputs),

at least one output port capable of being coupled to at least one second other preselected computation node included in the plurality of interconnected preselected computation nodes (Figure 1 and Page 427, see below), and

a router to direct the output data from the at least one output port to the at least one preselected second other computation node (Page 427, second column, paragraph 2. Any consumer can receive any data from the interconnection network, thus all processors are capable of sending data to any other processor).

11. As per Claim 39, Requa teaches: The method of claim 37, wherein at least one of the plurality of groups of instructions is a basic block (Page 426, first column third paragraph).

12. As per Claim 43, Requa teaches: The method of claim 37, wherein loading the group of instructions into a frame of buffers the plurality of interconnected preselected computation nodes includes:

sending at least two instructions selected from the group of instructions from an instruction sequencer to a selected computation node included in the plurality of interconnected preselected computation nodes for storage in a store (Page 426, Column 1, Paragraph 2, each instruction is sent out to a selected processor/functional unit/node).

13. As per Claim 44, Requa teaches: The method of claim 37, wherein executing the subset of instructions loaded into the frame of spanning buffers as each one of the instructions in the subset of instructions receives all necessary associated operands for execution includes:

matching at least one instruction selected from the group of instructions with at least one operand received from an other computation node included in the plurality of interconnected preselected computation nodes (Page 427, second column, second paragraph, where any consumer (processor) can receive any data, and instructions waiting to execute wait for results from the processor before executing).

14. As per Claim 46, Requa teaches: The method of claim 37, further comprising concurrently assigning another group of instructions selected from the plurality of groups of instructions to another or the same plurality of interconnected preselected computation nodes for concurrent execution using one or more other frames of buffers spanning the another or same plurality of interconnected preselected computation nodes (Page 436, Second Column, Second and Third paragraphs. Requa discusses

that blocks can overlap each other as long as they do not need results from each other. Additionally, in the third paragraph, Requa discusses that the PDF architecture is capable of supporting multiple program executions simultaneously, which would also read on the limitation):

wherein the two groups of instructions are capable of concurrent execution (Page 436, Second Column, Second and Third paragraphs. Requa discusses that blocks can overlap each other as long as they do not need results from each other. Additionally, in the third paragraph, Requa discusses that the PDF architecture is capable of supporting multiple program executions simultaneously, which would also read on the limitation).

15. As per Claim 47, Requa teaches: An article comprising a machine-accessible medium having machine executable instructions stored therein, configured to enable a machine to:

loading a subset of a group of instructions selected from a plurality of groups of instructions partitioned from a program to a frame of buffers spanning a plurality of interconnected preselected computation nodes (Page 426, first column, second paragraph, also see Page 433, "The PDF Block Processor". Specifically, see Page 434, Figure 9. the Instruction List, as it is used by all processors, and is a buffer, it is a buffer which spans all the connected nodes/processor

to execute the subset of instructions (Page 433, first column, third paragraph, an instruction waits for input operands, then is executed), wherein the group of instructions is assigned to be executed by the plurality of interconnected preselected computation nodes (Page 426, first column, second paragraph, the blocks are sent to processors).

16. As per Claim 48, Requa teaches: The article of claim 47, wherein partitioning the program into the plurality of groups of instructions is performed by a compiler (Page 429, first column, “PDF Architecture”).

17. As per Claim 50, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes information, which when accessed by the machine, results in the machine performing:

statically assigning all of the plurality of groups of instructions for execution (Page 432, see Figure 8, and column 1, paragraph 2).

18. As per Claim 52, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes instructions configured to enable the machine to:

generate a wakeup token to reserve an output data channel to connect selected computation nodes included in the plurality of interconnected preselected computation nodes (Page 427, section column, second paragraph).

19. As per Claim 53, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes instructions configured to enable the machine:

to repeat said loading until the entire group of instructions are executed (Page 433, in order to execute a block, this clearly has to occur), and

to detect execution termination of the group of instructions including an output having architecturally visible data (Page 433, second column first paragraph, a flag is set when an execution is done, also see Page 430, second paragraph); and

committing the architecturally visible data to a register file (Page 430, second paragraph).

20. As per Claim 54, Requa teaches: The article of claim 47, wherein the machine-accessible medium further includes instructions configured to enable the machine to repeat said loading until the entire group of instructions are executed (Page 433, in order to execute a block, this clearly has to occur), and to detect execution termination of the group of instructions including an output having architecturally visible data (Page 433, second column first paragraph, a flag is set when an execution is done, also see Page 430, second paragraph); and

committing the architecturally visible data to a memory (Page 430, second paragraph).

21. As per Claim 55, Requa teaches: The article of claim 47, wherein the machine-accessible medium further instructions configured to enable the machine to route an output datum arising from executing one of the subset of instructions to a consumer node included in the plurality of interconnected preselected computation nodes, wherein

the address of the consumer node is included in a token associated with at least one instruction included in the subset of instructions (Page 429, second column, second paragraph).

22. As per Claim 56, Requa teaches: The method of claim 37 further comprising repeating said loading and executing until the entire group of instructions have been executed (Page 433, in order to execute a block, this clearly has to occur).

23. As per Claim 57, Requa teaches: An apparatus, comprising:

a processor (Page 427, Figure 1, the Scalar, Memory, or SIMD Processor); and

storage medium coupled to the processor and having first instructions stored therein to be executed by the processor (Page 427, Main memory), wherein the first instructions are configured to

partition a program into a plurality of groups of second instructions (Page 426, first column, second paragraph, instructions are grouped into blocks);

assign a group of second instructions selected from the plurality of groups of second instructions partitioned from the program to a plurality of interconnected preselected computation nodes (Page 426, first column, second paragraph, the blocks are sent to processors); and

causing a subset of the second instructions of the assigned group of second instructions to be loaded into a frame of buffers spanning the plurality of interconnected preselected computation nodes having been assigned the group of second instructions

(Page 426, first column, second paragraph, also see Page 433, “The PDF Block Processor”. Specifically, see Page 434, Figure 9. the Instruction List, as it is used by all processors, and is a buffer, it is a buffer which spans all the connected nodes/processor), wherein the subset of second instructions are executed as each one of the instructions in the subset of second instructions loaded into the frame of spanning buffers receives all necessary associated operands for execution (Page 433, first column, third paragraph, an instruction waits for input operands, then is executed. Also see Page 435, Instruction Issue section. When the instruction in the list has all required operands, it is sent to the instruction-issue section, and then a processor).

24. As per Claim 58, Requa teaches: The apparatus of claim 57, wherein at least one of the plurality of groups of second instructions is a selected of one a basic block, a hyperblock or a superblock (Page 426, first column third paragraph).

25. As per Claim 59, Requa teaches: An apparatus, comprising:

computing resource including an execution unit configured to execute instructions (Page 427, Figure 1, the Scalar, Memory, or SIMD Processor); and

interconnect resource coupled to the computing resource to enable the apparatus to be a member of a group of interconnected computation nodes preselected to execute a group of instructions by successively executing subgroups of the group of instructions (Page 427, Second Column, Second paragraph, all processors/nodes are connected to each other through an interconnection network. Page 433 discloses successive execution of parts of a block);

wherein the interconnect resource includes:

at least one input port capable of being coupled to at least a first other preselected computation node included in the plurality of interconnected preselected computation nodes, the input port to receive input data (Page 427, Second Column, Second paragraph, every node is connected to every other node, requiring an input port),

a first store coupled to the at least one input port to store the input data (Page 427, Second Column, Second paragraph, the FIFO queue),

a second store coupled to the execution unit, the second store to receive and store at least one instruction of a subgroup, the second store being a part of a frame of buffers spanning the plurality of interconnected preselected computation nodes to store a subgroup of instructions (Page 426, first column, second paragraph, also see Page 433, "The PDF Block Processor". Specifically, see Page 434, Figure 9. the Instruction List, as it is used by all processors, and is a buffer, it is a buffer which spans all the connected nodes/processor),

an instruction wakeup unit to match the input data to the at least one stored instruction (PDF, paragraph 1, the operand source fields are modified as data comes in),

at least one output port coupled to the execution unit and capable of being coupled to at least one second other preselected computation node included in the plurality of interconnected preselected computation nodes (Page 427, Second Column, Second paragraph, every node is connected to every other node, requiring an output port), and

a router to direct an output data of the execution unit from the at least one output port to the at least one preselected other computation node (Page 427, the FIFO queue also acts as a kind of router).

26. As per Claim 60, Requa teaches: A system, comprising:

a plurality of interconnected computing nodes configured to be pre-selectable to cooperatively execute a group of instructions (Page 427, Figure 1, the Scalar, Memory, or SIMD Processors);

wherein each of the interconnect computing nodes includes:

computing resource including an execution unit configured to execute instructions (Page 427, Figure 1, the Scalar, Memory, or SIMD Processor); and

interconnect resource coupled to the computing resource to enable the computing node to cooperate with the other interconnected computation nodes to execute the group of instructions by successively executing subgroups of the group of instructions (Page 427, Second Column, Second paragraph, all processors/nodes are connected to each other through an interconnection network. Page 433 discloses successive execution of parts of a block);

wherein the interconnect resource includes:

at least one input port capable of being coupled to at least a first other preselected computation node including in the plurality of interconnected preselected computation nodes, the input port to receive input data (Page 427, Second Column, Second paragraph, every node is connected to every other node, requiring an input port),

a first store coupled to the at least one input port to store the input data (Page 427, Second Column, Second paragraph, the FIFO queue),

a second store coupled to the execution unit, the second store to receive and store at least one instruction of a subgroup, the second store being a part of a frame of buffers spanning the plurality of interconnected preselected computation nodes to store a subgroup of instructions (Page 426, first column, second paragraph, also see Page 433, "The PDF Block Processor". Specifically, see Page 434, Figure 9. the Instruction List, as it is used by all processors, and is a buffer, it is a buffer which spans all the connected nodes/processor),

an instruction wakeup unit to match the input data to the at least one stored instruction (PDF, paragraph 1, the operand source fields are modified as data comes in),

at least one output port coupled to the execution unit and capable of being coupled to at least one second other preselected computation node included in the plurality of interconnected preselected computation nodes (Page 427, Second Column, Second paragraph, every node is connected to every other node, requiring an output port), and a router to direct an output data of the execution unit from the at least one output port to the at least one preselected second other computation node (Page 427, the FIFO queue also acts as a kind of router).

### *Claim Rejections - 35 USC § 103*

27. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

28. Claims 40-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Requa, in view of Official Notice.

29. As per Claim 40, Requa teaches: The method of claim 37, but fails to teach:

wherein at least one of the plurality of groups of instructions is a hyperblock.

Requa teaches of a system which uses basic blocks, but does not teach that the groups may be hyperblocks. However, Examiner is taking Official Notice that one of ordinary skill in the art would be capable and motivated to use hyperblocks in lieu of basic blocks, to take advantage of the ability to have multiple exits from a block (While Applicant has not provided a definition of a hyperblock, Examiner has found it to represent a block with one entrance and potentially (but not necessarily) more than one exit).

30. As per Claim 41, Requa teaches: The method of claim 37, but fails to teach:

wherein at least one of the plurality of groups of instructions is a superblock.

Requa teaches of a system which uses basic blocks, but does not teach that the groups may be superblocks. However, Examiner is taking Official Notice that one of ordinary skill in the art would be capable and motivated to use superblocks in lieu of basic blocks, to take advantage of the ability to have multiple exits from a block (While Applicant has not provided a definition of a superblock, Examiner has found it to represent a block with one entrance and potentially (but not necessarily) more than one exit, however, Examiner is unclear how a superblock is different from a hyperblock).

31. Claims 42 and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Requa, in view of Fisher.

32. As per Claim 42, Requa teaches: The method of claim 37, but fails to teach:

wherein at least one of the plurality of groups of instructions is an instruction trace constructed by a hardware trace construction unit at run time.

While Requa teaches the method as disclosed in Claim 37, Requa does not teach about traces, or a trace construction unit to construct such a trace. However, Fisher teaches of Trace Scheduling, where the basic blocks used by Requa are compacted, and instead use traces (Page 462, Section D). The advantage to this compaction method using traces allows for more efficient parallel code, done in a manner far more efficient than previous methods (Abstract). Given this advantage, one of ordinary skill in the art would have been motivated to use these traces in place of the basic blocks as taught by Requa to further increase the efficiency of the system.

33. As per Claim 49, Requa teaches: The article of claim 47, but fails to teach:

wherein partitioning the program into the plurality of groups of instructions is performed by a run-time trace mapper.

While Requa teaches the article as disclosed in Claim 47, Requa does not teach that the partitioning of the program is done by a trace mapper. However, Fisher teaches of Trace Scheduling, where the basic blocks used by Requa are compacted, and instead use traces (Page 462, Section D), created and optimized by a scheduler (Page 482, second column, second paragraph). The advantage to this compaction method using traces allows for more efficient parallel code, done in a manner far more efficient than previous methods (Abstract). Given this advantage, one of ordinary skill in the art would have been motivated to use these traces in place of the basic blocks as taught by Requa to further increase the efficiency of the system.

### ***Response to Arguments***

34. Applicant has argued that because Requa teaches issuing each instruction of a group of instructions to a processor to execute, that Requa cannot teach "loading a subset of instructions of the assigned group of instructions into a frame of buffers spanning the plurality of interconnected preselected computation nodes having been assigned the group of instructions", along with executing them when their operands are

available. However, Examiner disagrees, as the instructions are stored somewhere before they are sent to processors, and Requa teaches that instructions cannot be executed until all of their operands are available (and this is an inherency anyway). Specifically, Examiner notes the Instruction List in the PDF processor which holds a portion of the block of instructions (and is arranged as a series of buffers, which spans all of the nodes), which sends out instructions to the processor, and receives more instructions from the block, until the block is executed. For these reasons, Examiner maintains the rejection over Requa.

35. If the Applicant wishes, the Examiner would be open to conducting an interview at the Applicant's request to discuss ways to bring the case potentially into allowability. While Examiner does not have any specific suggestions at this time, an elaboration of the buffers, or an explanation of what the current claims are doing, which Requa does not, could help in either differentiating the inventions, or helping to find a limitation that would make them patentably distinct, but as of the current time, Examiner does not see any patentable distinctions between Requa and the current invention, as they appear to perform the same function, with substantially identical hardware to perform the said same functions.

### *Conclusion*

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ROBERT E. FENNEMA whose telephone number is (571)272-2748. The examiner can normally be reached on Monday-Friday, 8:30-6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eddie P Chan/
Supervisory Patent Examiner, Art Unit 2183

Robert E Fennema
Examiner
Art Unit 2183

RF